Explaining Explainers: Necessity and Sufficiency in Tabular Data

Prithwijit Chowdhury
Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, GA 30332-0250
pchowdhury6@gatech.edu

Mohit Prabhushankar
Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, GA 30332-0250
mohit.p@gatech.edu

Ghassan AlRegib
Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, GA 30332-0250
alregib@gatech.edu

Abstract

In recent days, ML classifiers trained on tabular data are used to make efficient and fast decisions for various decision-making tasks. The lack of transparency in the decision-making processes of these models have led to the emergence of EXplainable AI (XAI). However, discrepancies exist among XAI programs, raising concerns about their accuracy. The notion of what an “important" and “relevant" feature is, is different for different explanation strategies. Thus grounding them using theoretically backed ideas of necessity and sufficiency can prove to be a reliable way to increase their trustworthiness. We propose a novel approach to quantify these two concepts in order to provide a means to explore which explanation method might be suitable for tasks involving the implementation of sparse high dimensional tabular datasets. Moreover, our global necessity and sufficiency scores aim to help experts to correlate their domain knowledge with our findings and also allow an extra basis for evaluation of the results provided by popular local explanation methods like LIME and SHAP.

1 Introduction

In high-stakes decision-making tasks, machine learning (ML) classifiers trained on tabular data are extensively utilized. While tabular data, due to its structured nature and analytical advantages, allows for organized and systematic data processing, ML classifiers provide faster and more efficient decision-making as compared to manual approaches [14]. However, the process by which these models make their predictions are generally hidden or uninterpretable to the end users. The field of XAI overcomes these challenges by providing human interpretable analysis in the form of transparent model explanations, intuitive visualizations, or understandable feature importance scores [3].

Among the various XAI techniques available, two types of popular explanation methods are: attribution-based and counterfactual-based. Attribution-based explanations entail determining the features that have the greatest impact on a model’s predictions [1, 7]. Methods like feature importance by approximation using linear models[21], partial dependence graphs [13], and Shapley [23] values are used to arrive at these explanations. In XAI, counterfactual based explanations entail creating

Table Representation Learning Workshop at NeurIPS 2023

alternative hypotheses to account for a model's predictions. In order to investigate "what-if" possibilities, these explanations alter the input attributes and track how the model's output changes [25]. Diverse Counterfactual Explanations (DiCE) [15], a common methodology for creating counterfactual explanations, uses optimization techniques to produce a variety of counterfactual examples that are similar to the original instance but have different outcomes.

However in spite of all the growing interest in XAI, there are discrepancies across well-known XAI programs, raising concerns about their accuracy in particular situations [16, 12, 11, 20]. Despite variations in the type of output and how it is produced, a convincing explanation should always satisfy the two desirable properties of necessity (*is a feature value necessary for generating the output of the model?*) and sufficiency (*is the feature value sufficient for generating the output of the model?*). Inclusion of necessity and sufficiency metrics in explanations thus increases trust in explainability approaches and hence, the underlying model results. Essential components are identified by necessity, while sufficiency ensures comprehensive justification. Current methods which formulate these two metrics either employee a causal model, which is difficult to obtain, or fail (or become computationally intensive) on sparse, imbalanced high dimensional tabular datasets which are common in domains like geophysics [17] and medicine [19]. Thus, following the framework established by [11], we introduce a direct perturbation based technique, which leverages the output of the model itself to guide our interventions, to generate counterfactuals and a novel method to calculate sufficiency. We provide a detailed analysis of how necessary or sufficient the important features, ranked by local explanation methods like LIME and SHAP, are in generating a particular prediction by the model. We find that in particular cases, a feature ranked by LIME or SHAP as more important than others is not so in terms of it's necessity and sufficiency.

To summarize, our work provides:

1. A novel model-centric method to calculate sufficiency scores for a feature set of a context (data-space) on which the model is being trained upon, allowing us to work with low density and high dimensional complex tabular data, without a causal map.

2. A global importance score for each feature set, based on their necessity and sufficiency to allow comparison with domain knowledge in order to solidify our trust in the method and also to act as the grounds to explain our following analysis of other attribution based methods.

3. A detailed analysis of the necessity and sufficiency of the important features ranked by the popular local explanation methods: LIME and SHAP. We raise and answer the two questions: ***Are the important features always necessary?*** and ***Are the important features sufficient enough?***

2 Background & related works

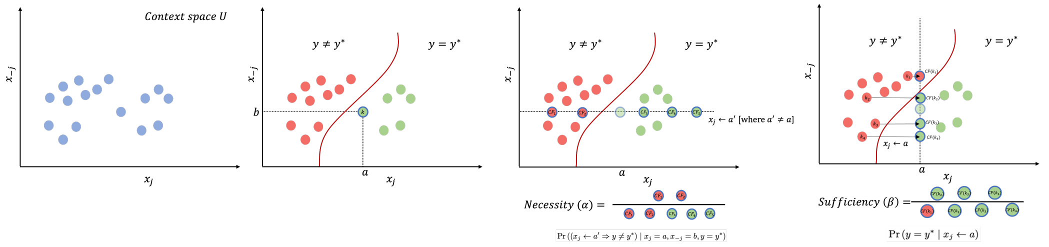
Necessity and sufficiency are concepts that have been extensively explored in philosophy, encompassing various logical, probabilistic, and causal interpretations. ([24]) defines necessity and sufficiency as follows:

Necessary condition: A condition **A** is said to be necessary for a condition **B**, iff the falsity of **A** guarantees the falsity of **B**. ***Example:****"Air is necessary for human life."*

Sufficient condition: A condition **A** is said to be sufficient for a condition **B**, iff the truth of **A** guarantees the truth of **B**. ***Example:****"Being a mammal is sufficient to have a spine."*

In propositional logic, we say that x is a sufficient condition for y iff $x \to y$, and x is a necessary condition for y iff $y \to x$.

These two concepts from causal analysis represent the characteristics that one would naturally anticipate the real cause of an event to display [18]. Recently, it has been claimed in a number of publications that model predictions can be explained using these same metrics of necessity and sufficiency. [11] determines the necessity and sufficiency scores for tabular data using [9]'s actual causality theory where they argued that for most realistic ML models, an ideal explanation is impractical.[27] offers an alternative approach for assessing requirement and sufficiency over subsets of features. [4] presents necessity and sufficiency as a novel feature attribution method for explaining text classifiers. [8] proposes a method to extract the concepts of sufficiency and necessity from probabilistic causal models. Our algorithm currently deals with non causal explanations in tabular datasets.



(a) A distribution of datapoints in Context space U

(b) A binary classifier is fit on the datapoints

(c) Calculating the necessity of feature x_j

(d) Calculating the sufficiency of feature x_j

Figure 1: This Figure explains how necessity and sufficiency calculation for a feature, in a classifier fitted distribution space, works. (a) Shows the distribution in the Context space U. This distribution is then fitted to a binary classifier model in (b) which separates the space based on the class prediction of $y = y^*$ & $y \neq y^*$. (c) and (d) captures the diagrammatic representation of how intervention is done on the feature x_j to a' and a respectively, to calculate the conditional probability α and β based on the change in prediction results.

In their work, [11] have provided the concept of partial explanations that relaxes the necessity and sufficiency conditions to consider the fraction of contexts over which these conditions are valid. Partial explanations are characterized by two metrics. The conditional probability metrics (α) and (β) respectively captures the extent to which a subset of feature values is either *necessary* or *sufficiency* to cause the model's (original) output.

Suppose there is a distribution of datapoints in a context space U (Fig.1(a)), where x_j is a subset of feature values relative to the set of context. x_{-j} represents the values of all variables except x_j. A binary classifier is fitted on the context space and a *reference datapoint* k is chosen, whose model output is $y = y^*$ (Fig.1(b)). The initial conditions are: For this datapoint, the desired feature x_j is equal to a. While the subset of all other features (x_{-j}) from the context U is valued at b. Now x_j is intervened and set to **any** a' [where $a' \neq a$] in (Fig.1(c)). The chances of the model output changing to something other than it's initial state of y^* when the intervention is carried out can thus be represented by:

$$\alpha = \Pr\left(\left(x_j \leftarrow a' \Rightarrow y \neq y^*\right) \mid x_j = a, x_{-j} = b, y = y^*\right) \tag{1}$$

Now to calculate β we intervene the value of x_j in different datapoints in the context space U in Fig.1(a) to our target value α (from reference point k in Fig.1(b). Thus, the conditional probability of sufficiency is the chances of the intervention, to the target value, leading to the target outcome $y = y^*$.

$$\beta = \Pr\left(y = y^* \mid x_j \leftarrow a\right) \tag{2}$$

Both probabilities (α, β), combined called the *goodness* of an explanation, are over the set of contexts U. When both $\alpha = 1$ and $\beta = 1$, $\alpha = 1$ captures that $x_j = a$ is a necessary cause of $y = y^*$ and $\beta = 1$ captures that $x_j = a$ is a sufficient cause of $y = y^*$. In other words, if the subset of feature values $x_j = a$ is an actual cause of the outcome $y = y^*$ with high probability, it is considered to be a good explanation for a model's output.

To quantify these goodness metrics [11] suggest a scoring system comprising a ***necessity score*** and ***sufficiency score***. Considering $y^* = f\left(x_j = a, x_{-j} = b\right)$ is the output of a classifier f for input x. they calculate the ***necessity score*** of a feature value $x_j = a$ for the model output y^*, by generating counterfactual explanations, but restricting it such that only x_j can be changed. The fraction of times changing x_j leads to the production of a valid DICE or Wachter [26] counterfactual example indicates that the extent to which $x_j = a$ is necessary for the current model output y^*.

They adopt the reverse approach for their sufficiency condition. Fixing x_j to its original value, they generated DICE and Wachter Counterfactuals (CFs) by letting all other features vary their values. The difference between the fraction of unique CFs generated using all features and the fraction of unique CFs generated while keeping x_j constant is their ***sufficiency score***.

However high-dimensional data poses a challenge known as the curse of dimensionality [6, 22]. As the number of dimensions increases, the distribution space grows exponentially, resulting in sparse data. This sparsity can make it difficult to generate meaningful counterfactuals (CFs) using DICE, as there might not be enough instances that match the desired conditions. As the number of dimensions in the data increases, the computational complexity of generating DICE counterfactuals grows significantly as well. Each additional dimension adds to the combinatorial explosion of possibilities, making it computationally expensive and time-consuming to explore all potential counterfactuals.

Furthermore finding erroneous correlations is more likely when there are several dimensions. It can be challenging to pinpoint the precise qualities that are causally responsible for the outcome in high-dimensional datasets because several aspects may be associated with one another. The identification of these causal effects might be made more difficult by the inclusion of confounding variables or hidden factors. Hence it is difficult to come up with a causal structure or model for these datasets, especially without prior knowledge of the domain we are dealing with. Hence the applications of [8]) in these conditions become very limited as well.

3 Method

For a counterfactual to be valid and unique, it must fulfil the three conditions of **1. causality** :*it should capture a causal relationship between input features and the model's output*, **2. plausibility**: *it should propose alternative instances that are plausible or realistic in the context of the problem domain* and **3. proximity**: *it should be close or similar to the original instance in terms of relevant features* [26]. As one starts dealing with complex high dimensional data, maintaining all of these conditions become increasingly challenging and thus the option to create a set of valid counterfactual becomes more and more constrained and limited.

Our method is a model agnostic (*flexible when it comes to model selection*), post-hoc (*calculations are done after the model has been trained*) scoring algorithm, which utilizes the predictions generated by the ML model itself, without further approximation. Contrary from DICE counterfactuals, which aims to generate unique counterfactuals, we operationalize Eq. (1) and Eq. (2), and directly calculate both necessity and sufficiency by intervening on a chosen feature (x_j) and causing perturbations on the observed datapoints, thus generating these pseudo-counterfactuals which don't necessarily have to be valid and unique or computationally intensive. Also these two equations ((1) and (2)) presume that individual feature are independent of each other, allowing any feature to be altered without affecting other features. This allows our algorithm to be functional without a causal model.

3.1 Counterfactual generation method:

For our algorithm we propose a forward "pseudo-counterfactual" generation method, where we directly make perturbations on the selected datapoint, without considering a target prediction. Suppose we have a trained binary classification model, $\mathbf{F}$ (fit on a set of context of tabular data, $\mathbf{U}$). We select an instance, $\mathbf{k}$ with feature vector $\mathbf{x} = [x_1, x_2, x_3, \ldots, x_n]$ from our context $\mathbf{U}$. Our intention is to generate counterfactual example(s) of this instance $\mathbf{k}$, $\{CF_1(k), CF_2(k), \ldots, CF_n(k)\}$, by introducing fine-grained variations on a single feature value (x_j) from the vector space $\mathbf{x}$, keeping all other feature sets constant. A unique counterfactual $CF_i(k)$ can thus be represented in the feature vector space as:

$$CF_i(k) = [x_1, x_2, \ldots, x_{j-1}, x_j^{(i)}, x_{j+1}, \ldots, x_n] \tag{3}$$

Here $x_j^{(i)}$ represents the perturbed value of feature x_j for counterfactual $CF_i(k)$, and all other features are kept the same as their values in $\mathbf{k}$.These newly generated set of counterfactuals are then provided for inference to the trained model $\mathbf{F}$ to obtain it's response.

3.2 Necessity score:

Eq. (1) can be deduced simply as the following argument: if we can change the model's output (from $y = y^*$ to $y \neq y^*$) by changing $\mathbf{x_j}$, it means that the $\mathbf{x_j}$ features' values are necessary to generate the model's original output ($y = y^*$). Hence we can conclude that for a chosen instance $\mathbf{k}$, the fraction of times a change in $\mathbf{x_j}$, while keeping all other features ($\mathbf{x_j}$) constant, leads to a change in $\mathbf{F}$'s output for the new set of n number of counterfactuals $\{CF_1(k), CF_2(k), \ldots, CF_n(k)\}$ is our desired necessity score for feature $\mathbf{x_j}$ (Fig. 1 c). This scoring is averaged over multiple ($\mathbf{N}$) separate instances in the context space $\mathbf{U}$.

$$\text{Necessity } = \frac{\sum^{N}\sum_{i}^{n}\mathbf{1}\left(CF_i(N) \mid x_j \neq a, y \neq y^*\right)}{\mathrm{n} * N} \tag{4}$$

where the model $\mathbf{F}$'s response for the initial instance $\mathbf{k}$ was $y = y^*$.

3.3 Sufficiency score:

Similarly as above, Eq (2) can be explained in the following way: if intervening $\mathbf{x_j}$ to a certain value of a leads us to a targeted output by the model (from $y \neq y^*$ to $y = y^*$), then we can conclude that $\mathbf{x_j}$ is sufficient to generate the model's desired output ($y = y^*$).To begin formulating our equation for sufficiency score, we first need to select our intervention value for $\mathbf{x_j}$ and desired outcome of our model $\mathbf{F}$. To do this, we choose a reference datapoint $\mathbf{r}$ from our context space $\mathbf{U}$ (Fig. 1 b). The value of feature $\mathbf{x_j}$ of $\mathbf{r}$ (suppose: $x_j = a$) becomes our intervention value for the corresponding feature and the model response to $\mathbf{r}$ is our desired/targetted outcome. (suppose: $y = \mathbf{F}(r) = y^*$).

Now, to draw our sample space to do our intervention for Eq. (2) we select a set of instances $\mathbf{K}$, which produces the undesired model response ($y \neq y^*$) from our test observational set. This is done to ensure that during intervention the newly generated counterfactual does not become too much of an outlier for our context U. For each $\mathbf{k} \in \mathbf{K}$, we intervene the value of $\mathbf{x_j}$ to a (Fig. 1 d).

Thus, the sufficiency score of feature $\mathbf{x_j}$ for reference point $\mathbf{r}$ can be computed as the fraction of counterfactuals ($CF(k)$) generated from the set $\mathbf{K}$, producing the targted outcome ($y = y^*$) from an undesired one ($y \neq y^*$) when $\mathbf{x_j}$ is intervened to value a ($x_j \leftarrow a$). This scoring is averaged over multiple ($\mathbf{R}$) references in the context space $\mathbf{U}$.

$$\text{Sufficiency } = \frac{\sum^{R}\sum^{K}\mathbf{1}\left(CF(k) \mid x_j \leftarrow a, y = y^*\right)}{\mathrm{K} * R} \tag{5}$$

where $k \in K$ and the model response $\mathbf{F}(\forall \mathbf{k})$ was the undesired one ($y \neq y^*$).

Using these two scores we come up with our own global importance scoring metric based on overall necessity and sufficiency of a particular feature set in an effort to establish the reliability of our algorithm by anchoring it to domain knowledge. In the next set of experiments we evaluate the explanations provided by two attribution based feature importance methods, LIME and SHAP, in different setups.

4 Experiments

We conduct two separate sets of experiments using our scoring metrics. First is to generate a global necessity and sufficiency values for each of the high level (either non-categorical or dictated by domain knowledge) features present in the dataset. Next is to conduct a necessity and sufficiency analysis of the features obtained as results of LIME and SHAP explanation.

To begin our experiments, we need to select our classifier models to calculate our scores upon. We have used a Logistic Regression model, a Gaussian Naive Bayes, a Random Forest Classifier and a Voting Classifier (which considers the weight of the previous three models in a $1 : 5 : 1$ proportion). All these models are selected from the popular sklearn library available for basic machine learning tasks in python. We split our dataset into a $70 : 30$ ratio to provide our training set and testing set (on which the interventions will be carried out) respectively.

4.1 Global feature importance using necessity and sufficiency

To calculate this we follow the method provided in [17] to summarize global model behavior through aggregation of local responses.

$$\Gamma^{i}_{\text{global}} = \frac{1}{|\mathcal{U}|}\sum_{k \in \mathcal{U}} \left|\gamma^{i}(k)\right| \tag{6}$$

Where Γ and γ are the global and local (necessity/sufficiency) score respectively, for a feature i. k is an element in the context space $\mathcal{U}$. $\Gamma^{i}_{\text{global}}$ is calculates by averaging the local score γ^{i} over all elements of $\mathcal{U}$. For our experiments we sample 100 random datapoints for the test set to create our space $\mathcal{U}$.

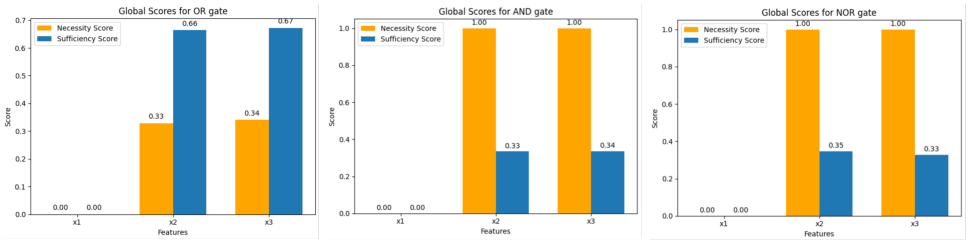
4.1.1 Validating our scores using a toy dataset

In order to validate the scores produced by our algorithm, we calculate the global necessity and sufficiency scores for datasets where we can estimate the necessity and sufficiency of each feature, manually, using logic. We synthetically generate 3 separate datasets with three features each. The first feature $x1$ is a float variable, while the second two features $x2$ and $x3$ are binary integers. The output Y is a logic AND, OR and NOR value of the two binary inputs ($x2$ and $x3$) respectively in each dataset.

The presence of $x1$ is to check if the model discovers any unnecessary correlation between it and the output.

Input Features			Ouptut Y		
x1	**x2**	**x3**	**AND**	**OR**	**NOR**
random_float_value	0	0	0	0	1
random_float_value	0	1	0	1	0
random_float_value	1	0	0	1	0
random_float_value	1	1	1	1	0

Table 1: Input Features and Output Y for the synthetic datasets: AND, OR and NOR.



(a) Global Scores for OR (b) Global Scores for AND (c) Global Scores for NOR

Figure 2: Global necessity and sufficiency scores for OR, AND and NOR dataset

Reference case: Any case with output: 1

In a logical AND: The intervention of either $x2$ or $x3$ to 0 will guarantee the output of Y to be false which is $3/3$ of the remaining cases, so $x2$ and $x3$ are **100% necessary.** Meanwhile, changing both $x2$ and $x3$ to 1 will guarantee the output of Y to be true (from false), which is $1/3$ of the remaining cases, so $x2$ and $x3$ are **33.33% sufficient.** Fig. 2a. shows the same results when calculated using our method.

In a logical OR: As reflected in Fig. 2b., the intervention of either $x2$ or $x3$ to 1 will guarantee the output of Y to be true (except case:(0,0)) which is $2/3$ of the remaining cases, so $x2$ and $x3$ are **66.66% sufficient.** Meanwhile, Changing both $x2$ and $x3$ to 0 will guarantee the output of Y to be false (from True), which is $1/3$ of the remaining cases, so $x2$ and $x3$ are **33.33% necessary.**

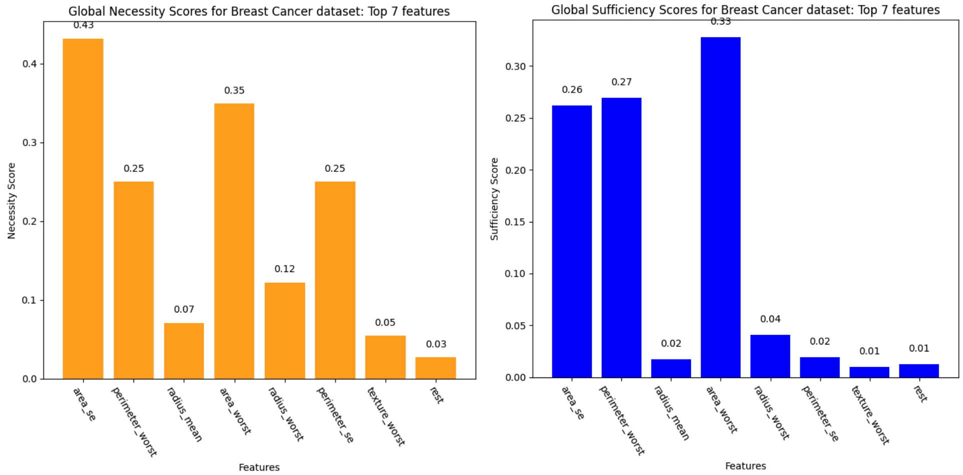
In a logical NOR: The intervention of both $x2$ and $x3$ to zero will only guarantee the output Y to be True (previously False). Which is possible only in $1/3$ of the remaining cases, so $x2$ and $x3$ should be **33.33% sufficient**. Meanwhile changing $x3$ and $x3$ to any values other than reference case (0,0) will guarantee the output Y to be False (from true). Which is $3/3$ of the remaining cases, so $x2$ and $x3$ should be **100% necessary**, which directly agrees with the findings in Fig. 2c.

4.2 Experiments on real datasets

We use three popular datasets used in ML based decision making tasks: AdultIncome [5], German-Credit[10]and Breast Cancer [28]. We use the default hyperparameters for LIME and SHAP for all our experiments. The main paper contains the global scoring and analysis on only the Breast Cancer dataset fitted on the Logistic Regression model. The main reason for picking this dataset was because of its sparsity (569 datapoints) and high-dimensionality (30 features). We get a training accuracy of 0.9497 and a test accuracy of 0.9825 on the above mentioned data-split. Results on the other datasets on all the models have been provided in the Supplementary Materials.

4.2.1 Global feature importance scores

Fig 3 a. & b. records the global necessity and sufficiency scores for the top 7 features calculated using Eq. 6 and a mean of the same scores for all the other features as "rest". It can be observed that for both necessity and sufficiency scoring, the top 7 as well as the top 3 features (*area se, perimeter worst and area worst*) are the same. These numbers start to make more sense when we correlate them with domain knowledge. According to [2] area, radius and perimeter are the features which determine the size and shape of the tumor (presence of cancer cells). While the features tagged with "worst" show the highest value recorded during treatment. Hence both *perimeter worst* and *area worst* having high scores checks out.



(a) Global necessary score

(b) Global sufficiency score

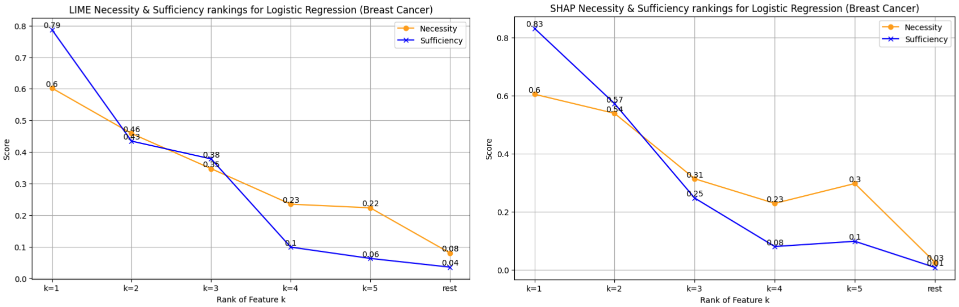
Figure 3: Global necessary and sufficiency scores for the breast cancer dataset.

4.2.2 Evaluating LIME and SHAP

In this section we use our scores to examine the necessity and sufficiency of the top features as declared by the two feature attribution methods, namely LIME and SHAP. This means that we use Eq. (4 & 5) to calculate the necessity and sufficiency scores of the k-th features in a LIME or SHAP explanation of a particular datapoint. Thus the k-th feature becomes out intervention feature (x_j) for that concerned datapoint. Provided the top "important" features identified based on these attribution methods, we compare the necessity and sufficiency scores of the the top k-th most important features ($k \in \{1, 2, 3, 4, 5\}$) with each other. Specifically we measure the average scores of these top k features, ranked by LIME and SHAP, over a span of multiple separate test datapoints. (N & R in Eq. (4 & 5)). We need to remember that all these scores are defined with respect to the feature value (a) and original output prediction (y) of each of these individual datapoints.

Practically no two same features are in the same importance ranking of two different datapoints, even by the same explanation method. Hence it raises the questions: ***Are the important features always necessary?*** and ***Are the important features sufficient?***. In an ideal settings, the most important feature ranked by an explanation should be both the most necessary and the most sufficient one.

The test was conducted for 40 random test samples and we can see from Fig. 4 that the pattern followed by both LIME and SHAP explanations are overall monotonic in nature. The most important feature ranked by both the local explanation method, tend to be the most necessary and sufficient for every datapoint. This can be verified by the complete dominance of the two most globally necessary and sufficient features (*perimeter worst* and *area worst*) in the Top 1 occurrence of both LIME and SHAP, recorded in Table 2. The trend showed by both of these graphs can be justified at any point by studying the occurrence of features at that point from Table 2 and weighing them by their global scores from Fig. 3.



(a) LIME - Necessity & Sufficiency

(b) SHAP - Necessity & Sufficiency

Figure 4: Necessity and sufficiency analysis of LIME and SHAP explanations of predictions for Gaussian NB model

	Explanation: LIME				Explanation: SHAP			
Feature Name	*Top 1 Occurrence*	*Top 2 Occurrence*	*Top 3 Occurrence*	*Top 5 Occurrence*	*Top 1 Occurrence*	*Top 2 Occurrence*	*Top 3 Occurrence*	*Top 5 Occurrence*
area_se	7	10	22	40	1	5	16	35
perimeter_worst	31	32	33	37	34	36	36	39
radius_mean	2	8	19	36	2	6	24	36
area_worst	0	23	34	34	0	30	34	35
radius_worst	0	0	3	31	0	0	2	33
perimeter_se	0	5	5	10	0	0	3	7
texture_worst	0	1	3	4	3	3	5	12
Total	40	79	119	192	40	80	120	197

Table 2: Top k occurrence for the Top 7 features of the Breast Cancer dataset as ranked by LIME during explanation of 40 test cases for model = Gaussian NB

5 Conclusion

The global scores help the user to ground their domain knowledge with our definitions of necessity and sufficiency. Establishing association between an expert's way of making a certain decision and the technique by which a model selects and chooses (necessary and sufficient) features to make a prediction will lead to a better understanding of the inner workings of the classifiers and ultimately will translate to higher levels of trust between the user and these ML models.

The necessity and sufficiency evaluation of LIME and SHAP in the main section as well as the appendix shows that the importance ranking of a feature by the explanation methods for different cases and different datasets differ based on the overall necessity or sufficiency of each individual feature and their ranking and presence, which often can be explained using domain knowledge and statistical analysis of the dataset itself. These evaluations can be further researched in future works to detect and control biases in datasets as well as feature selection methods.

In conclusion our study emphasises the value of employing several explanation techniques with a combination of different ml models, and also provides a proper evaluation method to determine which explanation technique and which model are most suitable for a particular task. When it comes to the precise topic of feature scoring or feature selection, the concept of "importance" seem to to be interpreted differently for different models and by different explanation methods. The temptation to view a particular explanation as a one-size-fits-all solution must be resisted by our scientific community. Overall, we think that in order to make a trustworthy and informed decision about the behavior of an ML model the employment of several explanation methods grounded by theoretical concepts and domain knowledge will only prove to be useful.

References

[1] Amina Adadi and Mohammed Berrada. Peeking inside the black-box: a survey on explainable artificial intelligence (xai). *IEEE access*, 6:52138–52160, 2018.

[2] Muhammet Fatih Ak. A comparative analysis of breast cancer detection and diagnosis using data visualization and machine learning applications. In *Healthcare*, volume 8, page 111. MDPI, 2020.

[3] Ghassan AlRegib and Mohit Prabhushankar. Explanatory paradigms in neural networks: Towards relevant and contextual explanations. *IEEE Signal Processing Magazine*, 39(4):59–72, 2022.

[4] Esma Balkir, Isar Nejadgholi, Kathleen C Fraser, and Svetlana Kiritchenko. Necessity and sufficiency for explaining text classifiers: A case study in hate speech detection. *arXiv preprint arXiv:2205.03302*, 2022.

[5] Barry Becker and Ronny Kohavi. Adult. UCI Machine Learning Repository, 1996. DOI: https://doi.org/10.24432/C5XW20.

[6] Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*, 2017.

[7] Filip Karlo Došilović, Mario Brčić, and Nikica Hlupić. Explainable artificial intelligence: A survey. In *2018 41st International convention on information and communication technology, electronics and microelectronics (MIPRO)*, pages 0210–0215. IEEE, 2018.

[8] Sainyam Galhotra, Romila Pradhan, and Babak Salimi. Explaining black-box algorithms using probabilistic contrastive counterfactuals. In *Proceedings of the 2021 International Conference on Management of Data*, pages 577–590, 2021.

[9] Joseph Y Halpern. *Actual causality*. MiT Press, 2016.

[10] Hans Hofmann. Statlog (German Credit Data). UCI Machine Learning Repository, 1994. DOI: https://doi.org/10.24432/C5NC77.

[11] Ramaravind Kommiya Mothilal, Divyat Mahajan, Chenhao Tan, and Amit Sharma. Towards unifying feature attribution and counterfactual explanations: Different means to the same end. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, pages 652–663, 2021.

[12] Satyapriya Krishna, Tessa Han, Alex Gu, Javin Pombra, Shahin Jabbari, Steven Wu, and Himabindu Lakkaraju. The disagreement problem in explainable machine learning: A practitioner's perspective. *arXiv preprint arXiv:2202.01602*, 2022.

[13] Scott M Lundberg, Gabriel G Erion, and Su-In Lee. Consistent individualized feature attribution for tree ensembles. *arXiv preprint arXiv:1802.03888*, 2018.

[14] Scott Mayer McKinney, Marcin Sieniek, Varun Godbole, Jonathan Godwin, Natasha Antropova, Hutan Ashrafian, Trevor Back, Mary Chesus, Greg S Corrado, Ara Darzi, et al. International evaluation of an ai system for breast cancer screening. *Nature*, 577(7788):89–94, 2020.

[15] Ramaravind K Mothilal, Amit Sharma, and Chenhao Tan. Explaining machine learning classifiers through diverse counterfactual explanations. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*, pages 607–617, 2020.

[16] W James Murdoch, Chandan Singh, Karl Kumbier, Reza Abbasi-Asl, and Bin Yu. Definitions, methods, and applications in interpretable machine learning. *Proceedings of the National Academy of Sciences*, 116(44):22071–22080, 2019.

[17] Ahmad Mustafa and Ghassan AlRegib. Explainable machine learning for hydrocarbon prospect risking. In *Second International Meeting for Applied Geoscience & Energy*, pages 1825–1829. Society of Exploration Geophysicists and American Association of Petroleum . . . , 2022.

[18] Judea Pearl. *Causality*. Cambridge university press, 2009.

[19] Mohit Prabhushankar, Kiran Kokilepersaud, Yash-yee Logan, Stephanie Trejo Corona, Ghassan AlRegib, and Charles Wykoff. Olives dataset: Ophthalmic labels for investigating visual eye semantics. *Advances in Neural Information Processing Systems*, 35:9201–9216, 2022.

[20] Yanou Ramon, David Martens, Foster Provost, and Theodoros Evgeniou. A comparison of instance-level counterfactual explanation algorithms for behavioral and textual data: Sedc, lime-c and shap-c. *Advances in Data Analysis and Classification*, 14:801–819, 2020.

[21] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

[22] Dylan Slack, Sophie Hilgard, Emily Jia, Sameer Singh, and Himabindu Lakkaraju. Fooling lime and shap: Adversarial attacks on post hoc explanation methods. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pages 180–186, 2020.

[23] Erik Štrumbelj and Igor Kononenko. Explaining prediction models and individual predictions with feature contributions. *Knowledge and information systems*, 41:647–665, 2014.

[24] Norman Swartz. The concepts of necessary conditions and sufficient conditions. *Department of Philosophy Simon Fraser University*, 1997.

[25] Arnaud Van Looveren, Janis Klaise, Giovanni Vacanti, and Oliver Cobb. Conditional generative models for counterfactual explanations. *arXiv preprint arXiv:2101.10123*, 2021.

[26] Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Harv. JL & Tech.*, 31:841, 2017.

[27] David S Watson, Limor Gultchin, Ankur Taly, and Luciano Floridi. Local explanations via necessity and sufficiency: Unifying theory and practice. In *Uncertainty in Artificial Intelligence*, pages 1382–1392. PMLR, 2021.

[28] Matjaz Zwitter and Milan Soklic. Breast Cancer. UCI Machine Learning Repository, 1988. DOI: https://doi.org/10.24432/C51P4M.

This appendix is divided in five sections. Appendix A provides the details about the three public datasets that have been used. Appendix B reports the global necessity and sufficiency scores for the top 5 features of the Adult Income and German Credit datasets. Appendix C and D reports the necessity and sufficiency analysis, respectively, of the LIME and SHAP explanations of Adult Income and German Credit datasets and appendix E lists the top k occurances of the top features as produces by the LIME and SHAP explanations.

A Dataset details

A.1 Breast Cancer Wisconsin (Diagnostic) Dataset

Origin: UCI Machine Learning Repository

Dataset Description: The Breast Cancer Wisconsin (Diagnostic) Dataset is a widely used dataset in the field of medical machine learning. It contains diagnostic information about breast cancer tumors and is used to classify tumors as benign (non-cancerous) or malignant (cancerous). The dataset is an essential resource for developing and evaluating models for breast cancer diagnosis.

Number of Datapoints: This dataset comprises a total of 569 datapoints.

Features: The dataset contains 30 features, which are numerical values representing various characteristics of cell nuclei from breast cancer biopsies. These features include mean, standard error, and worst (largest) values for attributes such as radius, texture, perimeter, area, smoothness, compactness, concavity, concave points, symmetry, and fractal dimension.

Data Types: All 30 features in this dataset are numerical.

Target Variable: The target variable is binary, with "M" representing malignant tumors and "B" representing benign tumors. The goal is to classify tumors into these two categories based on the features provided.

Training: Reported in the main Experiments section.

A.2 Adult Census Income Dataset

Origin: UCI Machine Learning Repository

Dataset Description: The Adult Census Income Dataset, often referred to as the "Census Income" or "Adult Income" dataset, is a widely used dataset for classification tasks, particularly for income prediction. It contains information related to census data and is used to predict whether an individual's income exceeds a certain threshold (typically $50,000 per year) based on various demographic and employment-related features.

Number of Datapoints: This dataset comprises a total of 48,842 datapoints.

Features: The dataset contains 14 features, including both categorical and numerical data types. Some of the features include age, workclass, education level, marital status, occupation, relationship status, race, gender, hours worked per week, native country, and more. These features provide a wide range of demographic and employment-related information about the individuals in the dataset.

Data Types: The dataset includes both categorical and numerical features.

Target Variable: The target variable in this dataset is binary, with two classes: "<=50K" (indicating that an individual's income is less than or equal to $50,000 per year) and ">50K" (indicating that an individual's income is greater than $50,000 per year). The goal is to predict an individual's income class based on the provided features

Training: A Logistic Regression model, A Gaussian Naive Bayes model, a Random Forest model and a Voting Classifier (which considers the weight of the previous three models in a $1:5:1$ proportion) was trained on this dataset on a $70:30$ split. The training accuracies achieved by these models are $0.8468, 0.9793, 0.7976$ and 0.8145 respectively. While the test accuracies were $0.8480, 0.8432, 0.8013$ and 0.8090 in the same order.

A.3 German Credit Dataset

Origin: UCI Machine Learning Repository

Dataset Description: The German Credit Dataset is a commonly used dataset in machine learning for credit risk analysis. It consists of 1,000 datapoints, with each datapoint representing an individual's credit application. The dataset aims to predict whether an applicant is a "good" or "bad" credit risk based on various features.

Features: This dataset comprises 20 features, including both categorical and numerical data types. Some of the features include the applicant's checking account status, duration of the credit, credit history, purpose of the credit, credit amount, savings account or bonds status, present employment duration, installment rate, personal status and sex, and more. These features provide information about the applicant's financial situation, employment, and personal details.

Target Variable: The target variable in this dataset is typically inferred as the "Credit Risk" label, which is binary, indicating whether an applicant is considered a good or bad credit risk. This label is not explicitly mentioned in the dataset, but it is a common task in credit risk analysis to predict this binary outcome.

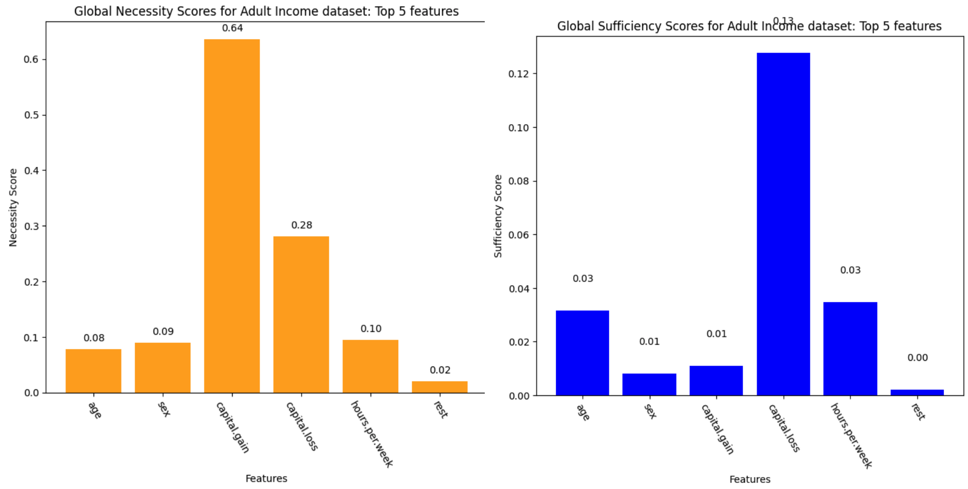
Training: A Logistic Regression model, A Gaussian Naive Bayes model, a Random Forest model and a Voting Classifier (which considers the weight of the previous three models in a $1 : 5 : 1$ proportion) was trained on this dataset on a $70 : 30$ split. The training accuracies achieved by these models are $0.6986, 1.00, 0.6740$ and 0.7123 respectively. While the test accuracies were $0.5924, 0.5860, 0.5414$ and 0.5605 in the same order.

B Global necessity and sufficiency scores

Fig. 5 and Fig. 6 provide the global necessity and sufficiency scores averaged across all models for the top 5 features in Adult Income datset and the 5 numerical features in German Credit dataset.

B.1 For Adult Income dataset

The top 5 features according to necessity and sufficiency are *age, sex, capital.gain, capital.loss* and *hours.per.week*, which also happen to be all of the numerical features present in the dataset. The categorical features were all one hot encoded during training and the overall mean global necessity and sufficiency scores are the lowest.



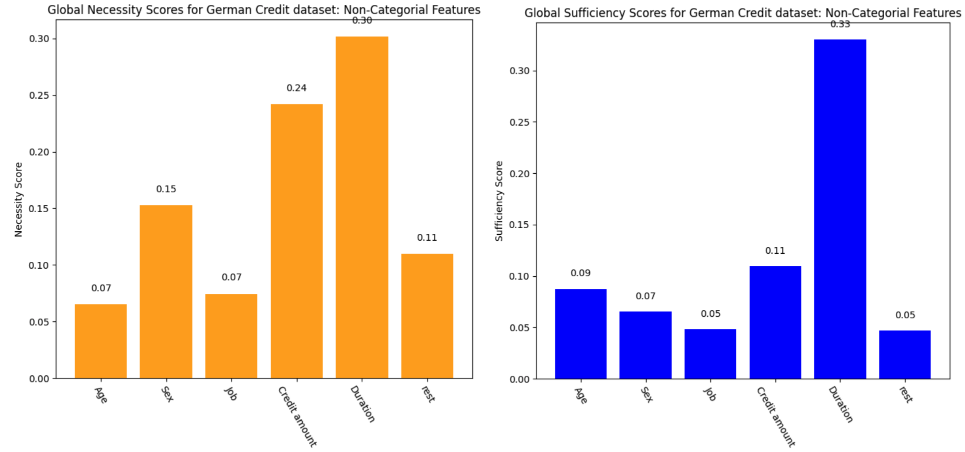
(a) Global Necessity

(b) Global Sufficiency

Figure 5: Global necessary and sufficiency scores for the Adult Income dataset

B.2 For German Credit dataset

The 5 features whose global necessity and sufficiency we have recorded for the German Credit dataset are *age, sex, job, capital amount* and *duration*, which are all of the numerical features present in the dataset. The categorical features were all one hot encoded during training, however their mean global scores have quite a high value than the other datset, hence each category of features have a overall higher necessity and sufficiency for model decision making.



(a) Global Necessity

(b) Global Sufficiency

Figure 6: Global necessary and sufficiency scores for the German Credit dataset

C Necessity evaluations of LIME and SHAP

In this section we use our scores to examine the necessity of the top features as declared by the two feature attribution methods, namely LIME and SHAP for Adult Income and German Credit dataset. Provided the top "important" features identified based on these attribution methods, we compare the necessity scores of the the top k-th most important features ($k \in \{1, 2, 3, 4, 5\}$) with each other. Specifically we measure the average scores of these top k features, ranked by LIME and SHAP, over a span of multiple seperate test datapoints. (N & R in Eq. (4 & 5)). We need to remember that all these scores are defined with respect to the feature value (a) and original output prediction (y) of each of these individual datapoints.

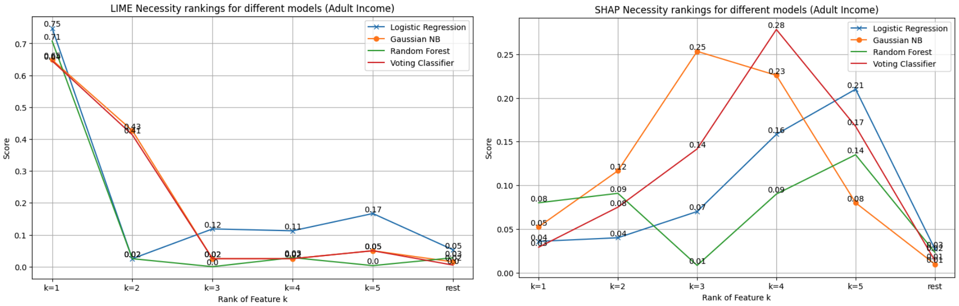
C.1 For Adult Income dataset

Fig 7 shows the Necessity analysis of LIME and SHAP explanations of predictions for models: Logistic Regression (*blue*), Gaussian NB(*orange*), Random Forest (*green*) and Voting Classifier (*red*) for Adult Income dataset.

The huge drop from $k = 1$ to $k = 2$ for necessity rankings of LIME in Fig. 7 a. is due to the fact that highest necessity scored feature *capital.gain* has all it's occurrence in 1st position, while the features with lower necessity scores show up in the next positions as displayed in Table 3.

The jump in necessity levels from $k = 2$ to the next positions for SHAP evaluation in Fig. 7 b. is due to the low occurrence of all the top 5 features in rank 1, with high occurances in each of the top 3 and 5 positions, shown in Table 3.

The trend showed by both of these graphs can be justified at any point by studying the occurrence of features at that point from Table 3 and weighing them by their global scores from Fig. 5.



(a) LIME - Necessity

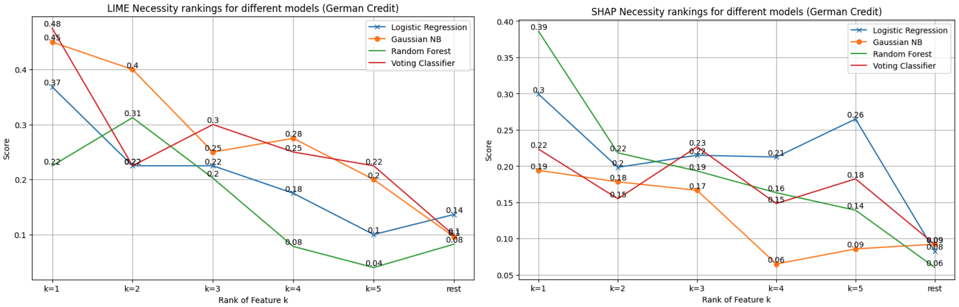
(b) SHAP - Necessity

Figure 7: Necessity analysis of LIME and SHAP explanations of predictions for models: Logistic Regression, Gaussian NB, Random Forest and Voting Classifier for Adult Income dataset.

C.2 For German Credit dataset

Fig 8 shows the Necessity analysis of LIME and SHAP explanations of predictions for models: Logistic Regression (*blue*), Gaussian NB(*orange*), Random Forest (*green*) and Voting Classifier (*red*) for German Credit dataset.

The rest of the trend shown by both of these graphs can be justified at any point by studying the occurrence of features at that point from Table 4 and weighing them by their global scores from Fig. 6.



(a) LIME - Necessity

(b) SHAP - Necessity

Figure 8: Necessity analysis of LIME and SHAP explanations of predictions for models: Logistic Regression, Gaussian NB, Random Forest and Voting Classifier for German Credit dataset.

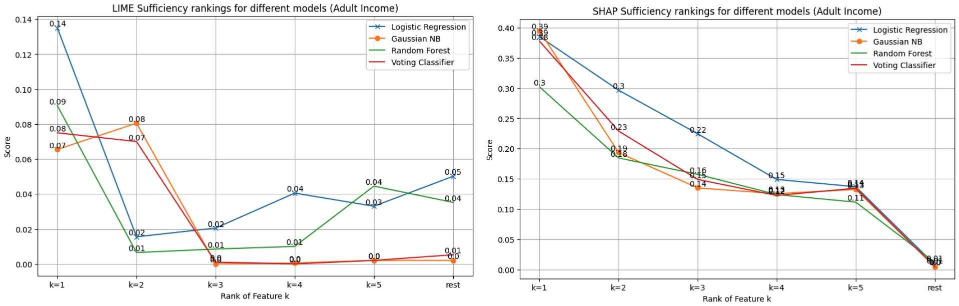
D Sufficiency evaluations of LIME and SHAP

In this section we use our scores to examine the sufficiency of the top features as declared by the two feature attribution methods, namely LIME and SHAP for Adult Income and German Credit dataset. Provided the top “important" features identified based on these attribution methods, we compare the sufficiency scores of the the top k-th most important features ($k \in \{1, 2, 3, 4, 5\}$) with each other. Specifically we measure the average scores of these top k features, ranked by LIME and SHAP, over a span of multiple seperate test datapoints. (N & R in Eq. (4 & 5)). We need to remember that all these scores are defined with respect to the feature value (a) and original output prediction (y) of each of these individual datapoints.

D.1 For Adult Income dataset

Fig 9 shows the Sufficiency analysis of LIME and SHAP explanations of predictions for models: Logistic Regression (*blue*), Gaussian NB(*orange*), Random Forest (*green*) and Voting Classifier (*red*) for Adult Income dataset.

The trend showed by both of these graphs can be justified at any point by studying the occurrence of features at that point from Table 3 and weighing them by their global scores from Fig. 5.



(a) LIME - Sufficiency

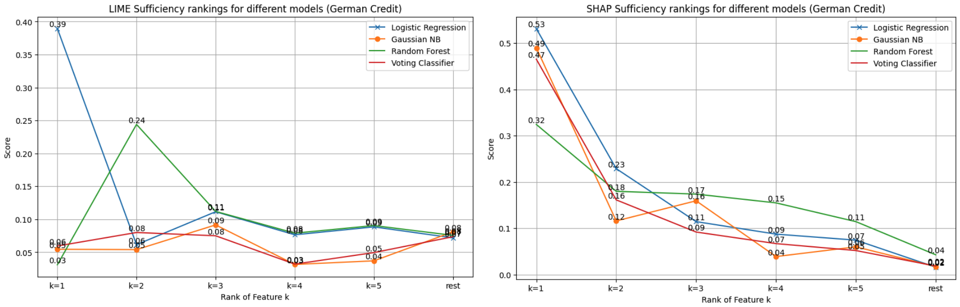
(b) SHAP - Sufficiency

Figure 9: Sufficiency analysis of LIME and SHAP explanations of predictions for models: Logistic Regression, Gaussian NB, Random Forest and Voting Classifier for Adult Income dataset.

D.2 For German Credit dataset

Fig 10 shows the Sufficiency analysis of LIME and SHAP explanations of predictions for models: Logistic Regression (*blue*), Gaussian NB(*orange*), Random Forest (*green*) and Voting Classifier (*red*) for German Credit dataset.

The trend showed by both of these graphs can be justified at any point by studying the occurrence of features at that point from Table 4 and weighing them by their global scores from Fig. 6.



(a) LIME - Sufficiency

(b) SHAP - Sufficiency

Figure 10: Sufficiency analysis of LIME and SHAP explanations of predictions for models: Logistic Regression, Gaussian NB, Random Forest and Voting Classifier for German Credit dataset.

E Top k occurrences of features ranked by LIME and SHAP

This section lists the top-k ($k = 1, 2, 3, 5$) occurrences of the top global necessary and sufficient features declared in Section B, for the trained Logistic Regression, Gaussian NB, Random Forest and Voting Classifier models.

Model		Explanation: LIME				Explanation: SHAP			
	Feature Name	*Top 1 Occurence*	*Top 2 Occurence*	*Top 3 Occurence*	*Top 5 Occurence*	*Top 1 Occurence*	*Top 2 Occurence*	*Top 3 Occurence*	*Top 5 Occurence*
Logistic Regression	**age**	0	0	0	0	0	3	7	16
	sex	0	0	0	0	3	9	12	23
	capital.gain	40	40	40	40	2	2	2	12
	capital.loss	0	0	3	27	3	4	4	4
	hours.per.week	0	0	0	0	4	4	7	10
Gaussian NB	**age**	0	0	0	0	0	0	2	8
	sex	0	0	0	0	0	0	1	5
	capital.gain	40	40	40	40	2	7	15	24
	capital.loss	0	40	40	40	4	4	4	5
	hours.per.week	0	0	0	0	0	0	1	3
Random Forest	**age**	0	0	0	0	14	19	24	27
	sex	0	0	0	0	0	0	2	3
	capital.gain	40	40	40	40	0	1	1	4
	capital.loss	0	0	1	16	2	2	2	3
	hours.per.week	0	0	0	0	3	5	9	14
Voting Classifier	**age**	0	0	0	0	0	2	5	13
	sex	0	0	0	0	0	1	3	9
	capital.gain	40	40	40	40	2	2	9	24
	capital.loss	0	40	40	40	4	4	4	5
	hours.per.week	0	0	0	0	0	1	4	6

Table 3: Top k occurrence for the Top 5 features of the Adult Income dataset as ranked by LIME during explanation of 40 test cases for model = Logistic Regression, Gaussian NB, Random Forest and Voting Classifier

Model		Explanation: LIME				Explanation: SHAP			
	Feature Name	*Top 1 Occurence*	*Top 2 Occurence*	*Top 3 Occurence*	*Top 5 Occurence*	*Top 1 Occurence*	*Top 2 Occurence*	*Top 3 Occurence*	*Top 5 Occurence*
Logistic Regression	**age**	0	0	4	12	3	7	19	23
	sex	0	0	7	31	7	13	24	40
	job	0	0	0	0	0	1	2	14
	credit amount	0	0	0	0	0	0	0	0
	duration	20	20	20	20	20	25	26	32
Gaussian NB	**age**	0	0	0	0	0	0	0	0
	sex	0	0	0	0	0	0	0	1
	job	0	0	0	0	0	0	0	1
	credit amount	0	0	10	0	3	3	10	17
	duration	0	0	0	0	9	15	20	28
Random Forest	**age**	0	0	0	0	0	1	3	15
	sex	0	0	0	1	1	4	5	9
	job	0	7	0	0	1	3	5	9
	credit amount	0	7	17	23	12	18	22	30
	duration	0	23	24	24	12	19	24	30
Voting Classifier	**age**	0	0	0	0	0	0	0	0
	sex	0	0	0	0	0	0	2	5
	job	0	0	0	0	0	0	0	0
	credit amount	0	0	0	0	3	3	4	11
	duration	0	0	0	0	8	16	21	28

Table 4: Top k occurrence for the Top 5 features of the German Credit dataset as ranked by LIME during explanation of 40 test cases for model = Logistic Regression, Gaussian NB, Random Forest and Voting Classifier